UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/721,630	11/25/2003	Thomas J. Dinger	LOT920030021US1	7718

45544   7590   06/24/2010
HOFFMAN WARNICK LLC
75 STATE ST
14TH FLOOR
ALBANY, NY 12207

EXAMINER
GISHNOCK, NIKOLAI A

ART UNIT	PAPER NUMBER
3715	

NOTIFICATION DATE	DELIVERY MODE
06/24/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTOCommunications@hoffmanwarnick.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/721,630
Filing Date: November 25, 2003
Appellant(s): DINGER ET AL.

Meghan Q. Toner
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 2/16/2010 appealing from the Office action mailed

9/24/2009.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by

name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings

which will directly affect or be directly affected by or have a bearing on the Board's decision in

the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

Claims 1-3 and 5

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of

amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in

the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection

to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the

appeal is taken (as modified by any advisory actions) is being maintained by the examiner

except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN

REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW

GROUNDS OF REJECTION."

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the

Appendix to the appellant's brief.

(8) Evidence Relied Upon

US 2003/0039949 A1   CAPPELLUCCI et al.   2-2003

Advanced Distributed Learning. ADL SCORM Version 1.3 Application Profile, Working Draft 0.9

(11-27-2002), pp. 5-18 through 5-22.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459

(1966), that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

1.   Determining the scope and contents of the prior art.
2.   Ascertaining the differences between the prior art and the claims at issue.
3.   Resolving the level of ordinary skill in the pertinent art.
4.   Considering objective evidence present in the application indicating obviousness
       or nonobviousness.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459

(1966), that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

1.   Determining the scope and contents of the prior art.
2.   Ascertaining the differences between the prior art and the claims at issue.
3.   Resolving the level of ordinary skill in the pertinent art.
4.   Considering objective evidence present in the application indicating obviousness
      or nonobviousness.

Claims 1-3 & 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Cappellucci et al. (US 2003/0039949 A1), hereinafter known as Cappellucci, in view of

Advanced Distributed Learning. *ADL SCORM Version 1.3 Application Profile, Working Draft 0.9*

[2002-11-27], hereinafter known as Advanced Distributed Learning.

Cappellucci teaches a computer-implemented method for performing branched rollup for

shared learning competencies in a learning environment, comprising: providing a hierarchical

tree corresponding to the learning environment (Para. 0053), wherein the hierarchical tree

includes a parent node, a first branch having a first child node and a first grandchild node, and a

second branch having a second child node and a second grandchild node (Para. 0054, Table 1;

also, Figure 4, Items M6.16, parent node, and subsequent nodes); providing a learning

competency in the learning environment that is shared by the first grandchild node and the

second grandchild node (two other {information element's} data category items can be

correlated against the same MLO {Master Learning Objective}, Para. 0056); performing an

information rollup using at least one computer device (performing a correlation query, a process

to find those information objects and elements that are correlated against a particular

information object or element; the system finds all information object or object correlated against

all MLOs which are state standards, correlated against lesson plans, and retrieves the

information objects or elements searched for, both in Para. 0072) of the first child node (an MLO

can be any node on the tree, Para. 0071) upon a change in state of the learning competency

(the system allows a user to modify an existing information object or element, Para. 0063; in the

event it is desirable to add information resources, each of these objects can be analyzed for

content and other metadata categories and correlated to the MLOs quickly and efficiently, Para.

0059); and performing an information rollup of the second child node after performing the

information rollup of the first child node (if no child MLOs are found the process can continue

where the system can search for all sibling MLOs of the initial MLOs found, and the system

tests to determine if any sibling MLOs were found, Para. 0074; also, Figure 8A, Items 822 &

824, and Figure 8B, Item 800); generating a control block for each of the first child node, the

second child node and the parent node prior to the first performing step (meta data populating

the data base of Para. 0076-0077 is generated when the information resource is input in the

system, Para. 0077), wherein the control block for the parent node indicates that the information

rollup of the first child node and the information rollup of second child node must both be

performed prior to performing the information rollup of the parent node (the correlation data

object {of the database identifying information resources} can include an MLO ID which

identifies the MLO that information object is correlated to, Para. 0076; two data category items

can be correlated against the same MLO, Para. 0056); and performing an information rollup of

the parent node only after performing the information rollup of the first child node and the

information rollup of the second child node, eliminating repeated rollups of the parent node (if no

sibling MLOs are found, the process can continue to where the system can search for all parent

MLOs of the initial MLOs found, the system tests to determine if any parent MLOs were found,

Para. 0074), and outputting the hierarchical tree (all information objects and elements available

from the system can be presented to a user at the user's computer or printed on a user's printer

or a system printer, Para. 0063) [Claim 1].

Cappellucci teaches a computerized system and a computer program product stored on a recordable medium for performing branched rollup for shared learning competencies in a learning environment, comprising: a list compilation system for generating a list of nodes that share a learning competency within a hierarchical tree corresponding to the learning environment (parsing system for establishing a correlation between information objects or elements and one or more MLOs, Para. 0066); a block generation system for generating control blocks for predecessors of the nodes in the list of nodes, wherein each of the control blocks identifies specific successors of the predecessors for which information rollups must be performed before information rollups of the predecessors can be performed (the system provides {correlation data object} tools which facilitate input of attributes of the information object or element, Para. 0077); and a node rollup system for processing the control blocks and performing the information rollups of the predecessors after performing the information rollups of the specific successors (correlation query process, Para. 0072-0074) [Claim 1].

Cappellucci teaches wherein the hierarchical tree comprises a parent node, a first branch having a first child node and a first grandchild node, and a second branch having a second child node and a second grandchild node (Para. 0054, Table 1; also, Figure 4, Items M6.16, parent node, and subsequent nodes) [Claim 1].

Cappellucci teaches wherein the learning competency is shared by the first grandchild node and the second grandchild node, wherein the first child node and the parent node are the predecessors of the first grandchild node, and wherein the second child node and the parent node are the predecessors of the second grandchild node (information objects are analyzed for content and other metadata categories correlated to the MLOs, Para. 0059; data category items can be correlated against the same MLO, Para. 0056; an MLO can be any node on the tree,

Para. 0071; it is inherent that MLOs can be parent nodes, child nodes, and first and second

grandchild nodes, as in Figure 4) [Claim 1].

Cappellucci teaches wherein the information rollup of the parent node is performed only

after the information rollup of the first child node and the information rollup of the second child

node are performed (if no child MLOs are found the process can continue where the system can

search for all sibling MLOs of the initial MLOs found, and the system tests to determine if any

sibling MLOs were found, Para. 0074; also, Figure 8A, Items 822 & 824, and Figure 8B, Item

800; if no sibling MLOs are found, the process can continue to where the system can search for

all parent MLOs of the initial MLOs found, the system tests to determine if any parent MLOs

were found, Para. 0074) [Claim 1].

Cappellucci teaches wherein the learning environment is implemented in a computerized

environment (Para. 0062) [Claim 1].

Cappellucci teaches wherein the information rollups of the first child node, the second

child node and the parent node are performed a maximum of one time for a change in state of

the learning competency (the system allows a user to modify an existing information object or

element, Para. 0063; in the event it is desirable to add information resources, each of these

objects can be analyzed for content and other metadata categories and correlated to the MLOs

quickly and efficiently, Para. 0059; a correlation query is a process to find those information

objects and elements that are correlated against a particular information object of element,

Para. 0072; also Figures 6 & 7; the correlation query is used to analyze the information objects

and elements when existing information objects or elements are modified), [Claim 1].

Cappellucci teaches wherein the list of nodes is generated, the control blocks are

generated and processed, and the information rollups are performed upon a change in state of

the learning competency (parsing, input of object meta data into database, and correlation query

performed {analysis for content and other meta data categories and correlation to the MLOs} in the event that information resources are added, Para. 0059, or an existing information object is modified, Para. 0063) [Claim 1].

What Cappellucci fails to explicitly teach is wherein the information rollups include communicating the change in state of the learning competency to node from which a predecessor depends [Claim 1]. However, Advanced Distributed Learning teaches that a Rollup is defined as the process of evaluating the Objective and Attempt Progress data for a set of child activities to determine the Objective and Attempt Progress data for the parent, and where the Rollup Rules define a set of rollup control rules for describing this processes [sic] (5.1.5 Rollup Rule Descriptions, page 5-18, first paragraph). Also, the example of Figure 5.1.5.6a, Rollup Rule Condition Illustration, further demonstrates communication of the state of the learning competency up a tree (Illustration 1 depicts a rollup rule that states all of the parent's (AA) children (AAA, AAB and AAC) activities have to be considered "satisfied", in order for its parent (AA) to be considered satisfied, Page 5-22, first paragraph). The rollup of Cappellucci would evaluate the progress data of a child node, according to the rule as taught by Advanced Distributed Learning, to determine the progress of the parent; this function causes the evaluation to be communicated to the parent. Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made, for the information rollup of Cappellucci to communicate the change in state of the learning competency of a grandchild node to its parent node, in the manner that information rollups are performed, as taught by Advanced Distributed Learning, in order to ensure that all the child learning competencies are satisfied before the parent node is completed, improving reliability of the course sequencing so that a child learning competency may not be skipped [Claim 1].

Cappellucci teaches analyzing the hierarchical tree to identify the second grandchild node as sharing the learning competency with the second grandchild node (information objects are analyzed for content and other metadata categories correlated to the MLOs, Para. 0059; data category items can be correlated against the same MLO, Para. 0056; an MLO can be any node on the tree, Para. 0071; it is inherent that MLOs can be first and second grandchild nodes); and adding the second grandchild node to a list of nodes (the system can include a separate database for identifying each type of information resource and new types of resource can be added as necessary, Para. 0077) prior to performing the information rollup of the first child node (as in Para 0074 and Figures 8A & 8B; in this case the rollup of a second grandchild node {a sibling node to a first grandchild} is performed prior to the child node {the parent of the grandchild nodes}, case being where the child is the initial MLO found) [Claim 2].

Cappellucci teaches consulting the list of nodes prior to performing the information rollup of the second child node (data structure can form part of a database that stores the meta data and is used in queries to find information objects and elements, Para. 0076; this meta data can be derived from available data when the information resource is input into the system, Para. 0077, the testing steps of Para. 0074 are consulting the correlation database of Para. 0076-0077 prior to each level of search) [Claim 3].

Cappellucci teaches processing the control block for the first child node prior to performing the information rollup of the first child node; processing the control block for the second child node prior to performing the information rollup of the second child node; and processing the control block for the parent node prior to performing the information rollup of the parent node (the system tests to determine if any child, sibling or parent MLOs were found, then continues to the next level, as in Para. 0074) [Claim 5].

(10) Response to Argument

The Dinger invention is a method of rolling up data in a hierarchical tree, comprising

providing a tree with a parent (or root) node, child nodes, and grandchild nodes (i.e., daughters

of the child nodes) with share a learning competency; generating a control block (understood to

be code) for each node; performing an information rollup of two child nodes "upon a change in

state of the learning competency, whereby the information rollups include communicating a

change in state of the learning competency to a node from which at least one of the grandchild

nodes depend (although unlimited by the claim language, it is easiest to assume that the state

change is communicated to the child nodes); and performing a roll-up of the parent node (from

the child nodes) only after rolling up the child nodes.

Cappellucci teaches parent, child and grandchild nodes at Figure 4, the nodes are

information objects (the control blocks) corresponding to what he calls MLOs or Master Learning

Objectives, which describe educational standards for state school systems, colleges, whatever

(Para. 0050-52). The MLOs are hierarchical (Para. 0070) and go all the way down to a bottom-

most node, such as Science/Physics/Optics/Lasers/Holograms "Student knows the difference

between constructive and destructive interference in holographic imaging". Cappelluci (at Para.

0074 and Figures 8A-B) flowchart a query for standards that correlate to a lesson plan, by

searching the child MLOs first, then the sibling MLOs, and finally the parent MLOs.

What Cappelluci is not explicit is on where information is necessarily communicated from

a grandchild node to a child node (showing that the information is "change of state" information

is not really needed here because the information is never used apart from initiating the

method). ADL SCORM manual describes the process on an information rollup of learning

conditions using rollup rules - the default rule being "All" for which all the children activities in a

tree must be satisfied in order for the parent activity to be satisfied. The rollup rule is evaluated

upon a change in activity completion status (page 5-22). ADL SCORM's "All" rollup rule wuld be used in Cappellucci's query to ensure course sequencing so that a required learning competency is not skipped.

Appellant's argument that Cappellucci and ADL SCORM fail to teach where the method of rolling up grandchildren first is "eliminating repeated rollups of the parent node" is not weighting in patentability because it is merely an additional use of steps that Cappellucci and ADL SCORM teach. Eliminating repeated rollups is not a step of the instant method. I also want to say it's inherent in Cappellucci's explicit order of steps, because the step of rolling up parent nodes is performed last and not repeated. Appellant's argument at page 6 that ADL SCORM simply ensures that a child node is not missed or skipped, but does not ensure that every child node is rolled up before a parent node is rolled up is incorrect because the words "in order to" show a cause and effect that all child nodes must be considered satisfied before their parent is satisifed. In a multi-level hierarchy like Cappellucci, a grandchild node may have children and grandchilren, it would be obvious that a grandchild node, possibly being the parent of child and grandchild nodes, must have their "Satisfied" state rolled up before its own, propagating the state information back to the root when an activity is completed. Thus, every child node by extension must be rolled "bottom-up" to the top level parent node to eliminate redundant rollups. Appellant's argument that generating a control block is something different than Cappellucci's metadata for querying and tracking MLOs is is not supported by a distinction; my position is that when Cappelluci defines search parameters to test and determine all MLOs correlated against the input information object and retrieves this information at Para. 0072-73, this is generating a control block to perform the rollup.

ADL Scorm 1.3 Working Draft states at page 5-18 through 5-22 that:

- **Rollup is defined as the process of evaluating the Objective and Attempt Progress data for a set of child activities to determine the Objective and Attempt Progress**

> data for the parent. The Rollup Rules define a set of rollup control rules for describing this processes;

- Rollup rules consist of a set of child activity conditions and a corresponding action or behavior that is performed if the set of conditions evaluates to true (if [child_activity_set condition_set] then [action/behavior]). Rollup rules are optional and should only be defined when needed by the content author;
- The Rollup Child Activity Set defines the set of children that are used during the processing of Rollup Rules. There are several mechanisms defined by the IMS Simple Sequencing Specification to enable the content author to define the child activities to use for evaluation: All (default value). If the combination of all of the children of the parent activity has a rollup condition that evaluates to true, then perform the specified action; Rollup Rule Illustration: In the first scenario, the rule defines a condition that states if "All" of AA's children are "Satisfied" then AA should be considered "Satisfied"; and
- Rollup Rule Condition Illustration: Illustration 1 depicts a rollup rule that states all of the parent's (AA) children (AAA, AAB and AAC) activities have to be considered "satisfied" in order to consider AA satisfied. Illustration 2 depicts a rollup rule that states only 2 of the children activities must have a status of "satisfied", in order for its parent (AA) to be considered satisfied. Illustration 3 depicts a rollup rule that is not satisfied. The Rollup Rule states that all of the children activities (AAA, AAB and AAC) must be satisfied in order for the parent activity (AA) to be satisfied. Child activity AAC status is Not Satisfied which cause the rollup rule to deem activity AA as not satisfied.

Cappelluci clearly teaches using at least one computing device in the rollup procedure (see Figure 5). Further, "generating a control block" is understood to be merely creating some computer code, which is also clearly performed by populating the database of Cappellucci (Para. 0076-77).

In response to the Appellant's arguments that neither the Cappellucci reference nor ADL SCORM white paper teach eliminating repeated rollups of the parent node by rolling up the state of a learning competency from grandchild nodes first, then child nodes, and lastly the parent node: ADL SCORM clearly states at page 5-22, 1st paragraph and at Figure 5.1.5.6a: Rollup Rule Condition Illustration, a rollup rule states that all of the parent's (AA) children (AAA, AAB, and AAC) activities have to be considered "satisfied" in order for the parent (AA) to be considered satisfied. As is explained above, it would have been obvious to one of ordinary skill in the art, at the time the invention was made, for the information rollup of Cappellucci to communicate the change in state of the learning competency of a grandchild node to its parent

node, in the manner that information rollups are performed, as taught by Advanced Distributed

Learning, in order to ensure that all the child learning competencies are satisfied before the

parent node is completed, improving reliability of the course sequencing so that a child learning

competency may not be skipped. Where Cappellucci states in Para. 0066 that: "One of the

benefits of the present system is that even with the manual process [of parsing the information

objects and elements for words and phrases likely to be descriptive of the information object or

element, searching the MLOs based upon the words and phrases found, and establishing a

correlation between each information object or element and one or more MLOs as a function

words or phrases found], the correlation step need only be performed once when the new

information is added as compared with the free form information model where each new

information object or element would have to be correlated with each existing information

resource of the system," this means that, because the correlations (i.e., links) between the

learning objectives (i.e., the MLOs) and the information objects and elements (resources) are

rolled up automatically, as described in Paras. 0074-76, the correlation step is performed only

once for each piece of new information. Cappellucci is thus understood to teach recursive

propagation of the learning competency information upwards through a hierarchical tree;

something that is well-known in the data structures and computer science arts. Further, in

response to appellant's arguments that Cappelluci fail to teach "generating control blocks" of the

instant invention, examiner's position is that such control blocks are merely program instructions

and data of the sort that any computer-implemented process creates within computer memory;

in this case, the links in the MLOs and resource information, as well as the data tree in ADL

SCORM created by he program in those inventions clearly demonstrates teachings of

generating control blocks for performing those functions. Appellant's arguments are thus

incorrect.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related

Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

6/19/2010

/Nikolai A Gishnock/

Examiner, Art Unit 3715

Conferees:

/XUAN M. THAI/

Supervisory Patent Examiner, Art Unit 3715

/Gene Kim/

Supervisory Patent Examiner, Art Unit 3711